**BODY WITH PACKAGE**

CREATE OR REPLACE PACKAGE salary\_pkg IS

g\_max\_sal\_raise CONSTANT NUMBER := 0.20;

FUNCTION validate\_raise

(

p\_old\_salary IN employees.salary%TYPE,

p\_new\_salary IN employees.salary%TYPE

)

RETURN BOOLEAN;

PROCEDURE update\_sal

(

p\_employee\_id IN employees.employee\_id%TYPE,

p\_new\_salary IN employees.salary%TYPE

);

END salary\_pkg;

/

CREATE OR REPLACE PACKAGE BODY salary\_pkg IS

FUNCTION validate\_raise

(

p\_old\_salary IN employees.salary%TYPE,

p\_new\_salary IN employees.salary%TYPE

)

RETURN BOOLEAN IS

BEGIN

IF p\_new\_salary > (p\_old\_salary \* (1 + g\_max\_sal\_raise)) THEN

RETURN FALSE;

ELSE

RETURN TRUE;

END IF;

END validate\_raise;

PROCEDURE update\_sal

(

p\_employee\_id IN employees.employee\_id%TYPE,

p\_new\_salary IN employees.salary%TYPE

) IS

v\_old\_salary employees.salary%TYPE;

BEGIN

SELECT salary

INTO v\_old\_salary

FROM employees

WHERE employee\_id = p\_employee\_id;

IF validate\_raise(v\_old\_salary, p\_new\_salary) THEN

UPDATE employees

SET salary = p\_new\_salary

WHERE employee\_id = p\_employee\_id;

ELSE

RAISE\_APPLICATION\_ERROR(-20001, 'Salary raise exceeds the maximum allowed.');

END IF;

END update\_sal;

END salary\_pkg;

/

BEGIN

salary\_pkg.update\_sal(100,25000);

END;

/

SELECT \* FROM EMPLOYEES;

